SQL
Cheat Sheet

Commands | Query | Joins
Operators | Functions

Frank Andrade

SQL Cheat Sheet

SQL is a language designed to manage and retrieve information from relational databases.

Table : Players

Columns →

id	Name	Age	Nationality	Weight
1	Lionel	32	ARG	72
2	Cristiano	34	PRT	83
3	Neymar	27	BRA	68

→ Records

Data Types

1.- Numeric:

a) Integers: BIT, TINYINT, SMALLINT, INT, BIGINT

b) Decimals: MONEY, DECIMAL

2.- Text: CHAR, VARCHAR, NCHAR, NVARCHAR, BINARY. VARBINARY

3.- Date: DATE, TIME, DATETIME, SMALLDATETIME

Commands

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. HAVING
6. ORDER BY
7. LIMIT

Querying a Table

1) Select all the columns from the table
```
SELECT * FROM Players;
```

2) Select column Name from the table
```
SELECT Name FROM Players;
```

3) Select column Name and Weight from the table
```
SELECT Name, Weight FROM Players;
```

4) Select id, Name in asecending order by Age
```
SELECT id, Name FROM Players
ORDER BY Age ASC;
```

5) Select id, Name in descending order by Age
```
SELECT id, Name FROM Players
ORDER BY Age DESC;
```

6) Select first 5 rows from the table
```
SELECT * FROM Players
LIMIT 5;
```

Filtering Data

Filtering on numeric columns

1) Select all the columns where Age is greater or equal to 32
```
SELECT * FROM Players
WHERE Age >= 32;
```

2) Select all the columns where Age is less than 32
```
SELECT * FROM Players
WHERE Age < 32;
```

3) Select all the columns where age is between 27 and 32
```
SELECT * FROM Players
WHERE Age BETWEEN 27 AND 32;
```

Filtering on text columns

4) Select all the 'ARG' records
```
SELECT * FROM Players
WHERE Nationality = 'ARG';
```

5) Select all the 'ARG' and 'PRT' records
```
SELECT * FROM Players
WHERE Nationality IN ('ARG','PRT');
```

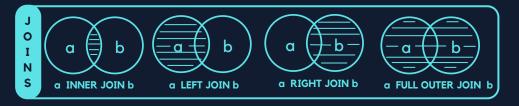**6)** Select all records where nationality starts with 'A' but doesn't end with 'T'
```
SELECT * FROM Players
WHERE Nationality LIKE 'A%' AND
Nationality NOT LIKE '%T';
```

Filtering multiple columns

7) Select all records with nationaility 'ARG' and age greater than 32
```
SELECT * FROM Players
WHERE Nationality ='ARG' AND
Age > 32;
```

8) Select all records with nationality 'ARG' or weights greater than 70
```
SELECT * FROM Players
WHERE Nationality ='ARG' OR
Weight > 70;
```

JOINS

a INNER JOIN b   a LEFT JOIN b   a RIGHT JOIN b   a FULL OUTER JOIN b

Filtering missing data

9) Select records where the column `Age` is null

```sql
SELECT * FROM Players
WHERE Age IS NULL;
```

10) Select records where the column `Age` is not null

```sql
SELECT * FROM Players
WHERE Age IS NOT NULL;
```

Aggregate Functions

Simple aggregations

1) Get total sum of ages among players

```sql
SELECT SUM(Age) FROM Players;
```

2) Get the average age of players

```sql
SELECT AVG(Age) FROM Players;
```

3) Get the maximum age in the table

```sql
SELECT MAX(Age) FROM Players;
```

4) Get the minimum age in the table

```sql
SELECT MIN(Age) FROM Players;
```

Grouping, filtering and sorting

5) Get sum of ages grouped by nationality

```sql
SELECT Nationality, SUM(Age)
FROM Players
GROUP BY Nationality;
```

6) Get the average age grouped by nationality

```sql
SELECT Nationality, AVG(Age)
FROM Players
GROUP BY Nationality;
```

7) Get the maximum age grouped by nationality

```sql
SELECT Nationality, MAX(Age)
FROM Players
GROUP BY Nationality;
```

8) Get the minimum age grouped by nationality

```sql
SELECT Nationality, MIN(Age)
FROM Players
GROUP BY Nationality;
```

9) Get the average age grouped by nationality in ascending order

```sql
SELECT Nationality,
       AVG(Age) AS AVG_AGE
FROM Players
GROUP BY Nationality
ORDER BY AVG_AGE ASC;
```

10) For Argentina and Portugal, get the average age grouped by nationality

```sql
SELECT Nationality, AVG(Age)
FROM Players
WHERE Nationality IN ('ARG','PRT')
GROUP BY Nationality;
```

11) Get the number of elements per nationality

```sql
SELECT Nationality, COUNT(Name)
FROM Players
GROUP BY Nationality;
```

More Coming Soon ...
(stay tuned!)